

TechShop

Task 1. Database Design:

1. Create the database named "TechShop"

```
mysql> create database TechShop;
Query OK, 1 row affected (0.03 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sakila |
| sys |
| techshop |
| world |
+-----+
7 rows in set (0.00 sec)
```

2. Define the schema for the Customers, Products, Orders, OrderDetails and Inventory tables based on the provided schema.

1. Customers:

• CustomerID (Primary Key) • FirstName • LastName • Email • Phone • Address

```
CREATE TABLE Customers (
    CustomerID INT PRIMARY KEY,
    FirstName VARCHAR(100),
    LastName VARCHAR(100),
    Email VARCHAR(100),
    Phone VARCHAR(20),
    Address VARCHAR(150)
);
```

```
mysql>
mysql> CREATE TABLE Customers (
  ->     CustomerID INT PRIMARY KEY,
  ->     FirstName VARCHAR(100),
  ->     LastName VARCHAR(100),
  ->     Email VARCHAR(100),
  ->     Phone VARCHAR(20),
  ->     Address VARCHAR(150)
  -> );
Query OK, 0 rows affected (0.04 sec)
```

```
mysql> desc Customers;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| CustomerID | int           | NO   | PRI | NULL    |       |
| FirstName  | varchar(100)  | YES  |     | NULL    |       |
| LastName   | varchar(100)  | YES  |     | NULL    |       |
| Email      | varchar(100)  | YES  |     | NULL    |       |
| Phone      | varchar(20)   | YES  |     | NULL    |       |
| Address    | varchar(150)  | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

2. Products: • ProductID (Primary Key) • ProductName • Description • Price

```
CREATE TABLE Products (
    ProductID INT PRIMARY KEY,
    ProductName VARCHAR(100),
    Description TEXT,
    Price DECIMAL(10, 2)
);
```

```
mysql> CREATE TABLE Products (
  ->     ProductID INT PRIMARY KEY,
  ->     ProductName VARCHAR(100),
  ->     Description TEXT,
  ->     Price DECIMAL(10, 2)
  -> );
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> desc Products;
```

Field	Type	Null	Key	Default	Extra
ProductID	int	NO	PRI	NULL	
ProductName	varchar(100)	YES		NULL	
Description	text	YES		NULL	
Price	decimal(10,2)	YES		NULL	

4 rows in set (0.00 sec)

3. Orders: • OrderID (Primary Key) • CustomerID (Foreign Key referencing Customers) • OrderDate • TotalAmount

```
CREATE TABLE Orders (
```

```
    OrderID INT PRIMARY KEY,
```

```
    CustomerID INT,
```

```
    OrderDate DATE,
```

```
    TotalAmount DECIMAL(10, 2)
```

```
);
```

```
mysql> CREATE TABLE Orders (
    ->     OrderID INT PRIMARY KEY,
    ->     CustomerID INT,
    ->     OrderDate DATE,
    ->     TotalAmount DECIMAL(10, 2)
    -> );
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> desc Orders;
```

Field	Type	Null	Key	Default	Extra
OrderID	int	NO	PRI	NULL	
CustomerID	int	YES	MUL	NULL	
OrderDate	date	YES		NULL	
TotalAmount	decimal(10,2)	YES		NULL	

4 rows in set (0.00 sec)

4. OrderDetails: • OrderDetailID (Primary Key) • OrderID (Foreign Key referencing Orders) • ProductID (Foreign Key referencing Products) • Quantity

```
CREATE TABLE OrderDetails (
    OrderDetailID INT PRIMARY KEY,
    OrderID INT,
    ProductID INT,
    Quantity INT
);
```

```
mysql> CREATE TABLE OrderDetails (
    ->     OrderDetailID INT PRIMARY KEY,
    ->     OrderID INT,
    ->     ProductID INT,
    ->     Quantity INT
    -> );
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> desc OrderDetails;
+-----+-----+-----+-----+-----+-----+
| Field          | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| OrderDetailID | int  | NO   | PRI | NULL    |       |
| OrderID       | int  | YES  | MUL | NULL    |       |
| ProductID     | int  | YES  | MUL | NULL    |       |
| Quantity      | int  | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)
```

5. Inventory • InventoryID (Primary Key) • ProductID (Foreign Key referencing Products) • QuantityInStock • LastStockUpdate

```
CREATE TABLE Inventory (
    InventoryID INT PRIMARY KEY,
    ProductID INT,
    QuantityInStock INT,
    LastStockUpdate DATE
);
```

```
mysql> CREATE TABLE Inventory (
->     InventoryID INT PRIMARY KEY,
->     ProductID INT,
->     QuantityInStock INT,
->     LastStockUpdate DATE
-> );
```

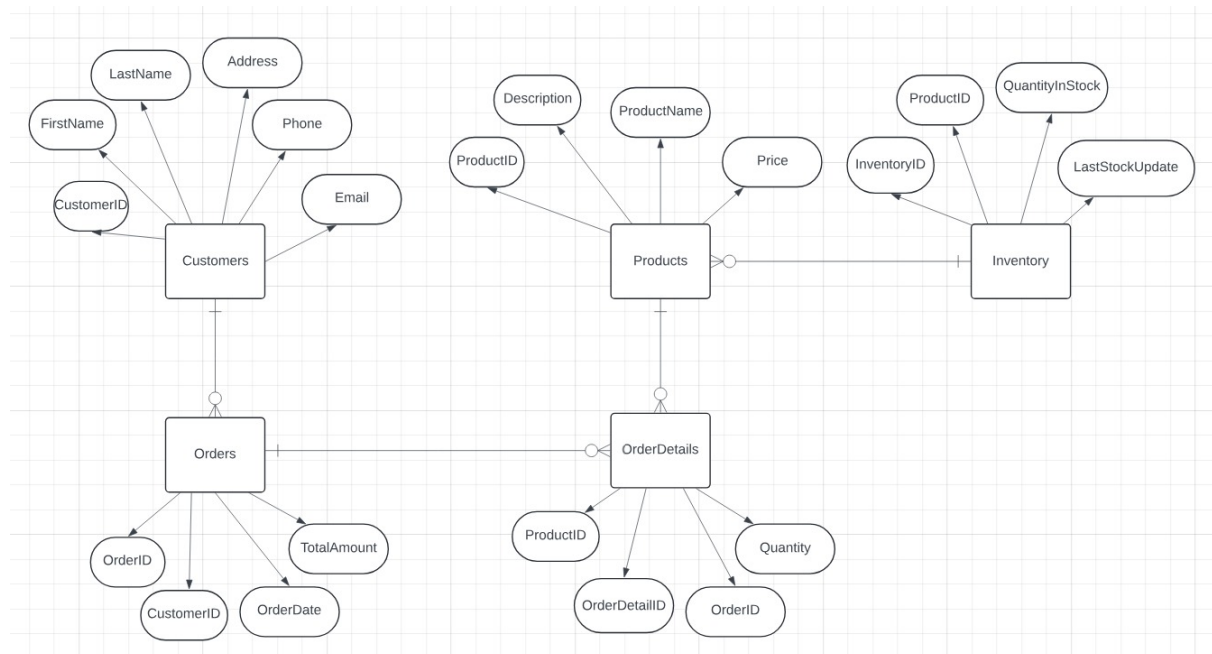
Query OK, 0 rows affected (0.03 sec)

```
mysql> desc inventory;
```

Field	Type	Null	Key	Default	Extra
InventoryID	int	NO	PRI	NULL	
ProductID	int	YES		NULL	
QuantityInStock	int	YES		NULL	
LastStockUpdate	date	YES		NULL	

4 rows in set (0.00 sec)

3. Create an ERD (Entity Relationship Diagram) for the database.



4. Create appropriate Primary Key and Foreign Key constraints for referential integrity.

-Adding Foreign Key constraint to Orders table

```
ALTER TABLE Orders
```

```
ADD CONSTRAINT fk_customer
```

```
FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID);
```

```
mysql> ALTER TABLE Orders
  -> ADD CONSTRAINT fk_customer
  -> FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID);
Query OK, 0 rows affected (0.13 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

-Adding Foreign Key constraints to OrderDetails table

ALTER TABLE OrderDetails

ADD CONSTRAINT fk_order

FOREIGN KEY (OrderID) REFERENCES Orders(OrderID);

```
mysql> ALTER TABLE OrderDetails
  -> ADD CONSTRAINT fk_order
  -> FOREIGN KEY (OrderID) REFERENCES Orders(OrderID);
Query OK, 0 rows affected (0.15 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

ALTER TABLE OrderDetails

ADD CONSTRAINT fk_product

FOREIGN KEY (ProductID) REFERENCES Products(ProductID);

```
mysql> ALTER TABLE OrderDetails
  -> ADD CONSTRAINT fk_product
  -> FOREIGN KEY (ProductID) REFERENCES Products(ProductID);
Query OK, 0 rows affected (0.17 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

-Adding Foreign Key constraint to Inventory table

ALTER TABLE Inventory

ADD CONSTRAINT fk_product_inventory

FOREIGN KEY (ProductID) REFERENCES Products(ProductID);

```
mysql> ALTER TABLE Inventory
  -> ADD CONSTRAINT fk_product_inventory
  -> FOREIGN KEY (ProductID) REFERENCES Products(ProductID);
Query OK, 0 rows affected (0.15 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

5. Insert at least 10 sample records into each of the following tables. a. Customers b. Products c. Orders d. OrderDetails e. Inventory

-Inserting sample data into Customers table

```
INSERT INTO Customers (CustomerID, FirstName, LastName, Email, Phone, Address)
VALUES
```

```
(1, 'Ravi', 'Kumar', 'ravi.kumar@gmail.com', '1234567890', '123 Main St, Vijayawada'),
(2, 'Sarala', 'Reddy', 'sarala.reddy@gmail.com', '0987654321', '456 Main St, Guntur'),
(3, 'Prasad', 'Rao', 'prasad.rao@gmail.com', '9876543210', '789 Main St, Visakhapatnam'),
(4, 'Anusha', 'Gowda', 'anusha.gowda@gmail.com', '0123456789', '101 Main St, Tirupati'),
(5, 'Rajesh', 'Babu', 'rajesh.babu@gmail.com', '8765432109', '222 Main St, Nellore'),
(6, 'Shobha', 'Chowdary', 'shobha.chowdary@gmail.com', '5432109876', '333 Main St, Kakinada'),
(7, 'Prakash', 'Naidu', 'prakash.naidu@gmail.com', '6543210987', '444 Main St, Rajahmundry'),
(8, 'Vijaya', 'Kumari', 'vijaya.kumari@gmail.com', '2345678901', '555 Main St, Amaravati'),
(9, 'Ram', 'Gopal', 'ram.gopal@gmail.com', '4321098765', '666 Main St, Vizianagaram'),
(10, 'Lakshmi', 'Devi', 'lakshmi.devi@gmail.com', '3456789012', '777 Main St, Anantapur');
```

```
mysql> INSERT INTO Customers (CustomerID, FirstName, LastName, Email, Phone, Address)
  -> VALUES
  -> (1, 'Ravi', 'Kumar', 'ravi.kumar@gmail.com', '1234567890', '123 Main St, Vijayawada'),
  -> (2, 'Sarala', 'Reddy', 'sarala.reddy@gmail.com', '0987654321', '456 Main St, Guntur'),
  -> (3, 'Prasad', 'Rao', 'prasad.rao@gmail.com', '9876543210', '789 Main St, Visakhapatnam'),
  -> (4, 'Anusha', 'Gowda', 'anusha.gowda@gmail.com', '0123456789', '101 Main St, Tirupati'),
  -> (5, 'Rajesh', 'Babu', 'rajesh.babu@gmail.com', '8765432109', '222 Main St, Nellore'),
  -> (6, 'Shobha', 'Chowdary', 'shobha.chowdary@gmail.com', '5432109876', '333 Main St, Kakinada'),
  -> (7, 'Prakash', 'Naidu', 'prakash.naidu@gmail.com', '6543210987', '444 Main St, Rajahmundry'),
  -> (8, 'Vijaya', 'Kumari', 'vijaya.kumari@gmail.com', '2345678901', '555 Main St, Amaravati'),
  -> (9, 'Ram', 'Gopal', 'ram.gopal@gmail.com', '4321098765', '666 Main St, Vizianagaram'),
  -> (10, 'Lakshmi', 'Devi', 'lakshmi.devi@gmail.com', '3456789012', '777 Main St, Anantapur');
Query OK, 10 rows affected (0.02 sec)
Records: 10 Duplicates: 0 Warnings: 0
```

-Inserting sample data into Products table

```
INSERT INTO Products (ProductID, ProductName, Description, Price)
```

```
VALUES
```

```
(1, 'HP Laptop', 'High performance laptop', 49999.99),  
(2, 'Smartphone', 'Latest smartphone model', 29999.99),  
(3, 'Tablet', 'Compact tablet', 14999.99),  
(4, 'Headphones', 'Wireless headphones', 9999.99),  
(5, 'Smartwatch', 'Fitness smartwatch', 7999.99),  
(6, 'Camera', 'DSLR camera', 34999.99),  
(7, 'TV', 'Smart LED TV', 54999.99),  
(8, 'Gaming Console', 'Next-gen gaming console', 19999.99),  
(9, 'Printer', 'Wireless printer', 9999.99),  
(10, 'Router', 'High-speed router', 4999.99);
```

```
mysql> INSERT INTO Products (ProductID, ProductName, Description, Price)  
-> VALUES  
-> (1, 'HP Laptop', 'High performance laptop', 49999.99),  
-> (2, 'Smartphone', 'Latest smartphone model', 29999.99),  
-> (3, 'Tablet', 'Compact tablet', 14999.99),  
-> (4, 'Headphones', 'Wireless headphones', 9999.99),  
-> (5, 'Smartwatch', 'Fitness smartwatch', 7999.99),  
-> (6, 'Camera', 'DSLR camera', 34999.99),  
-> (7, 'TV', 'Smart LED TV', 54999.99),  
-> (8, 'Gaming Console', 'Next-gen gaming console', 19999.99),  
-> (9, 'Printer', 'Wireless printer', 9999.99),  
-> (10, 'Router', 'High-speed router', 4999.99);  
Query OK, 10 rows affected (0.02 sec)  
Records: 10  Duplicates: 0  Warnings: 0
```

-Inserting sample data into Orders table

```
INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount)
```

```
VALUES
```

```
(1, 1, '2024-04-01', 49999.99),  
(2, 2, '2024-04-02', 29999.99),  
(3, 3, '2024-04-03', 14999.99),  
(4, 4, '2024-04-04', 9999.99),  
(5, 5, '2024-04-05', 7999.99),
```


(6, 6, '2024-04-06', 34999.99),
(7, 7, '2024-04-07', 54999.99),
(8, 8, '2024-04-08', 19999.99),
(9, 9, '2024-04-09', 9999.99),
(10, 10, '2024-04-10', 4999.99);

```
mysql> INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount)
-> VALUES
-> (1, 1, '2024-04-01', 49999.99),
-> (2, 2, '2024-04-02', 29999.99),
-> (3, 3, '2024-04-03', 14999.99),
-> (4, 4, '2024-04-04', 9999.99),
-> (5, 5, '2024-04-05', 7999.99),
-> (6, 6, '2024-04-06', 34999.99),
-> (7, 7, '2024-04-07', 54999.99),
-> (8, 8, '2024-04-08', 19999.99),
-> (9, 9, '2024-04-09', 9999.99),
-> (10, 10, '2024-04-10', 4999.99);
Query OK, 10 rows affected (0.02 sec)
Records: 10 Duplicates: 0 Warnings: 0
```

-Inserting sample data into OrderDetails table

INSERT INTO OrderDetails (OrderDetailID, OrderID, ProductID, Quantity)

VALUES

(1, 1, 1, 1),
(2, 2, 2, 1),
(3, 3, 3, 1),
(4, 4, 4, 1),
(5, 5, 5, 1),
(6, 6, 6, 1),
(7, 7, 7, 1),
(8, 8, 8, 1),
(9, 9, 9, 1),
(10, 10, 10, 1);

```
mysql> INSERT INTO OrderDetails (OrderDetailID, OrderID, ProductID, Quantity)
-> VALUES
-> (1, 1, 1, 1),
-> (2, 2, 2, 1),
-> (3, 3, 3, 1),
-> (4, 4, 4, 1),
-> (5, 5, 5, 1),
-> (6, 6, 6, 1),
-> (7, 7, 7, 1),
-> (8, 8, 8, 1),
-> (9, 9, 9, 1),
-> (10, 10, 10, 1);
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

-Inserting sample data into Inventory table

```
INSERT INTO Inventory (InventoryID, ProductID, QuantityInStock, LastStockUpdate)
```

```
VALUES
```

```
(1, 1, 10, '2024-03-01'),
(2, 2, 20, '2024-03-02'),
(3, 3, 15, '2024-03-03'),
(4, 4, 30, '2024-03-04'),
(5, 5, 25, '2024-03-05'),
(6, 6, 5, '2024-03-06'),
(7, 7, 8, '2024-03-07'),
(8, 8, 12, '2024-03-08'),
(9, 9, 18, '2024-03-09'),
(10, 10, 22, '2024-03-10');
```

```
mysql> INSERT INTO Inventory (InventoryID, ProductID, QuantityInStock, LastStockUpdate)
-> VALUES
-> (1, 1, 10, '2024-03-01'),
-> (2, 2, 20, '2024-03-02'),
-> (3, 3, 15, '2024-03-03'),
-> (4, 4, 30, '2024-03-04'),
-> (5, 5, 25, '2024-03-05'),
-> (6, 6, 5, '2024-03-06'),
-> (7, 7, 8, '2024-03-07'),
-> (8, 8, 12, '2024-03-08'),
-> (9, 9, 18, '2024-03-09'),
-> (10, 10, 22, '2024-03-10');
Query OK, 10 rows affected (0.02 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

Tasks 2. Select, Where, Between, AND, LIKE:

1. Write an SQL query to retrieve the names and emails of all customers.

Select FirstName, LastName, Email from Customers;

```
mysql> Select FirstName, LastName, Email from Customers;
```

FirstName	LastName	Email
Ravi	Kumar	ravi.kumar@gmail.com
Sarala	Reddy	sarala.reddy@gmail.com
Prasad	Rao	prasad.rao@gmail.com
Anusha	Gowda	anusha.gowda@gmail.com
Rajesh	Babu	rajesh.babu@gmail.com
Shobha	Chowdary	shobha.chowdary@gmail.com
Prakash	Naidu	prakash.naidu@gmail.com
Vijaya	Kumari	vijaya.kumari@gmail.com
Ram	Gopal	ram.gopal@gmail.com
Lakshmi	Devi	lakshmi.devi@gmail.com

```
10 rows in set (0.00 sec)
```

2. Write an SQL query to list all orders with their order dates and corresponding customer names.

Select o.OrderID, o.OrderDate, CONCAT(c.FirstName, ' ', c.LastName) AS CustomerName

FROM Orders AS o JOIN Customers AS c ON c.CustomerID = o.CustomerID;

```
mysql> Select o.OrderID, o.OrderDate, CONCAT(c.FirstName, ' ', c.LastName) AS CustomerName
-> FROM Orders AS o JOIN Customers AS c ON c.CustomerID = o.CustomerID;
```

OrderID	OrderDate	CustomerName
1	2024-04-01	Ravi Kumar
2	2024-04-02	Sarala Reddy
3	2024-04-03	Prasad Rao
4	2024-04-04	Anusha Gowda
5	2024-04-05	Rajesh Babu
6	2024-04-06	Shobha Chowdary
7	2024-04-07	Prakash Naidu
8	2024-04-08	Vijaya Kumari
9	2024-04-09	Ram Gopal
10	2024-04-10	Lakshmi Devi

```
10 rows in set (0.01 sec)
```

3. Write an SQL query to insert a new customer record into the "Customers" table. Include customer information such as name, email, and address.

INSERT INTO Customers(CustomerID, FirstName, LastName, Email, Phone, Address)

VALUES

(11, 'Ram', 'Reddy', 'ram@gmail.com', '1234123412', '123 Main St Vijayawada');

```
mysql> INSERT INTO Customers(CustomerID, FirstName, LastName, Email, Phone, Address)
-> VALUES
-> (11, 'Ram', 'Reddy', 'ram@gmail.com', '1234123412', '123 Main St Vijayawada');
Query OK, 1 row affected (0.01 sec)
```

4. Write an SQL query to update the prices of all electronic gadgets in the "Products" table by increasing them by 10%.

UPDATE Products

Set price = 1.1 * price;

```
mysql> UPDATE Products
-> Set price = 1.1 * price;
Query OK, 10 rows affected, 10 warnings (0.02 sec)
Rows matched: 10  Changed: 10  Warnings: 10

mysql> select * from Products;
+-----+-----+-----+-----+
| ProductID | ProductName | Description | Price |
+-----+-----+-----+-----+
| 1 | HP Laptop | High performance laptop | 54999.99 |
| 2 | Smartphone | Latest smartphone model | 32999.99 |
| 3 | Tablet | Compact tablet | 16499.99 |
| 4 | Headphones | Wireless headphones | 10999.99 |
| 5 | Smartwatch | Fitness smartwatch | 8799.99 |
| 6 | Camera | DSLR camera | 38499.99 |
| 7 | TV | Smart LED TV | 60499.99 |
| 8 | Gaming Console | Next-gen gaming console | 21999.99 |
| 9 | Printer | Wireless printer | 10999.99 |
| 10 | Router | High-speed router | 5499.99 |
+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

5. Write an SQL query to delete a specific order and its associated order details from the "Orders" and "OrderDetails" tables. Allow users to input the order ID as a parameter.

SET @order_id = 10;

DELETE FROM Orders WHERE OrderID = @order_id;

DELETE FROM OrderDetails WHERE OrderID = @order_id;

```
mysql> SET @order_id = 10;
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> DELETE FROM OrderDetails WHERE OrderID = @order_id;  
Query OK, 1 row affected (0.03 sec)  
  
mysql> DELETE FROM Orders WHERE OrderID = @order_id;  
Query OK, 1 row affected (0.01 sec)
```

6. Write an SQL query to insert a new order into the "Orders" table. Include the customer ID, order date, and any other necessary information.

```
INSERT INTO OrderDetails (OrderDetailID, OrderID, ProductID, Quantity)
```

```
VALUES (10, 10, 10, 3);
```

```
INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount)
```

```
VALUES (10, 1, '2024-04-10', 39999.99);
```

```
mysql> INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount)  
-> VALUES (10, 1, '2024-04-10', 39999.99);  
Query OK, 1 row affected (0.02 sec)
```

7. Write an SQL query to update the contact information (e.g., email and address) of a specific customer in the "Customers" table. Allow users to input the customer ID and new contact information.

```
SET @new_email = 'example@example.com';
```

```
SET @new_address = '466 Main St, Vizag';
```

```
SET @customer_id = 10;
```

```
mysql> SET @new_email = 'example@example.com';  
Query OK, 0 rows affected (0.00 sec)  
  
mysql> SET @new_address = '466 Main St, Vizag';  
Query OK, 0 rows affected (0.00 sec)  
  
mysql> SET @customer_id = 10;  
Query OK, 0 rows affected (0.00 sec)
```

```
UPDATE Customers
```

```
SET Email = @new_email, Address = @new_address
```

```
WHERE CustomerID = @customer_id;
```

```
mysql> UPDATE Customers
-> SET Email = @new_email, Address = @new_address
-> WHERE CustomerID = @customer_id;
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from customers;
```

CustomerID	FirstName	LastName	Email	Phone	Address
1	Ravi	Kumar	ravi.kumar@gmail.com	1234567890	123 Main St, Vijayawada
2	Sarala	Reddy	sarala.reddy@gmail.com	0987654321	456 Main St, Guntur
3	Prasad	Rao	prasad.rao@gmail.com	9876543210	789 Main St, Visakhapatnam
4	Anusha	Gowda	anusha.gowda@gmail.com	0123456789	101 Main St, Tirupati
5	Rajesh	Babu	rajesh.babu@gmail.com	8765432109	222 Main St, Nellore
6	Shobha	Chowdary	shobha.chowdary@gmail.com	5432109876	333 Main St, Kakinada
7	Prakash	Naidu	prakash.naidu@gmail.com	6543210987	444 Main St, Rajahmundry
8	Vijaya	Kumari	vijaya.kumari@gmail.com	2345678901	555 Main St, Amaravati
9	Ram	Gopal	ram.gopal@gmail.com	4321098765	666 Main St, Vizianagaram
10	Lakshmi	Devi	example@example.com	3456789012	466 Main St, Vizag
11	Ram	Reddy	ram@gmail.com	1234123412	123 Main St Vijayawada

```
11 rows in set (0.01 sec)
```

8. Write an SQL query to recalculate and update the total cost of each order in the "Orders" table based on the prices and quantities in the "OrderDetails" table.

UPDATE Orders o

SET TotalAmount = (

SELECT SUM(od.Quantity * p.Price)

FROM OrderDetails od

JOIN Products p ON od.ProductID = p.ProductID

WHERE od.OrderID = o.OrderID

)

WHERE o.OrderID IN (

SELECT OrderID

FROM OrderDetails

);

```
mysql> UPDATE Orders o
-> SET TotalAmount = (
->     SELECT SUM(od.Quantity * p.Price)
->     FROM OrderDetails od
->     JOIN Products p ON od.ProductID = p.ProductID
->     WHERE od.OrderID = o.OrderID
-> )
-> WHERE o.OrderID IN (
->     SELECT OrderID
->     FROM OrderDetails
-> );
Query OK, 9 rows affected (0.05 sec)
Rows matched: 9  Changed: 9  Warnings: 0
```

9. Write an SQL query to delete all orders and their associated order details for a specific customer from the "Orders" and "OrderDetails" tables. Allow users to input the customer ID as a parameter.

```
SET @customer_id = 8;
```

```
DELETE FROM OrderDetails WHERE OrderID IN (SELECT OrderID FROM Orders WHERE
CustomerID = @customer_id);
```

```
DELETE FROM Orders WHERE CustomerID = @customer_id;
```

```
mysql> SET @customer_id = 8;
Query OK, 0 rows affected (0.00 sec)

mysql> DELETE FROM OrderDetails WHERE OrderID IN (select OrderID from Orders WHERE CustomerID = @customer_id);
Query OK, 1 row affected (0.01 sec)

mysql> select * from orderdetails;
+-----+-----+-----+-----+
| OrderDetailID | OrderID | ProductID | Quantity |
+-----+-----+-----+-----+
| 1 | 1 | 1 | 1 |
| 2 | 2 | 2 | 1 |
| 3 | 3 | 3 | 1 |
| 4 | 4 | 4 | 1 |
| 5 | 5 | 5 | 1 |
| 6 | 6 | 6 | 1 |
| 7 | 7 | 7 | 1 |
| 9 | 9 | 9 | 1 |
+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

```
mysql> DELETE FROM Orders WHERE CustomerID = @customer_id;
Query OK, 1 row affected (0.01 sec)
```

```
mysql> select * from orders;
```

OrderID	CustomerID	OrderDate	TotalAmount
1	1	2024-04-01	54999.99
2	2	2024-04-02	32999.99
3	3	2024-04-03	16499.99
4	4	2024-04-04	10999.99
5	5	2024-04-05	8799.99
6	6	2024-04-06	38499.99
7	7	2024-04-07	60499.99
9	9	2024-04-09	10999.99
10	1	2024-04-10	39999.99

```
9 rows in set (0.00 sec)
```

10. Write an SQL query to insert a new electronic gadget product into the "Products" table, including product name, category, price, and any other relevant details.

```
INSERT INTO Products (ProductID, ProductName, Description, Price)
```

```
VALUES (11, 'Dell Laptop', 'High performance laptop', 49999.99);
```

```
mysql> INSERT INTO Products (ProductID, ProductName, Description, Price)
-> VALUES (11, 'Dell Laptop', 'High performance laptop', 49999.99);
Query OK, 1 row affected (0.02 sec)
```

```
mysql> select * from products;
```

ProductID	ProductName	Description	Price
1	HP Laptop	High performance laptop	54999.99
2	Smartphone	Latest smartphone model	32999.99
3	Tablet	Compact tablet	16499.99
4	Headphones	Wireless headphones	10999.99
5	Smartwatch	Fitness smartwatch	8799.99
6	Camera	DSLR camera	38499.99
7	TV	Smart LED TV	60499.99
8	Gaming Console	Next-gen gaming console	21999.99
9	Printer	Wireless printer	10999.99
10	Router	High-speed router	5499.99
11	Dell Laptop	High performance laptop	49999.99

```
11 rows in set (0.00 sec)
```

11. Write an SQL query to update the status of a specific order in the "Orders" table (e.g., from "Pending" to "Shipped"). Allow users to input the order ID and the new status.

```
ALTER TABLE Orders
```

```
ADD COLUMN Status VARCHAR(50);
```



```
mysql> ALTER TABLE Orders
  -> ADD COLUMN Status VARCHAR(50);
Query OK, 0 rows affected (0.31 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> select * from orders;
```

OrderID	CustomerID	OrderDate	TotalAmount	Status
1	1	2024-04-01	54999.99	NULL
2	2	2024-04-02	32999.99	NULL
3	3	2024-04-03	16499.99	NULL
4	4	2024-04-04	10999.99	NULL
5	5	2024-04-05	8799.99	NULL
6	6	2024-04-06	38499.99	NULL
7	7	2024-04-07	60499.99	NULL
9	9	2024-04-09	10999.99	NULL
10	1	2024-04-10	39999.99	NULL

```
9 rows in set (0.00 sec)
```

UPDATE Orders

SET Status = @new_status

WHERE OrderID = @order_id;

```
mysql> SET @new_status = 'shipped';
Query OK, 0 rows affected (0.00 sec)

mysql> SET @order_id = 9;
Query OK, 0 rows affected (0.00 sec)

mysql> UPDATE orders
  -> SET status = @new_status
  -> WHERE orderID = @order_id;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select * from orders;
```

OrderID	CustomerID	OrderDate	TotalAmount	Status
1	1	2024-04-01	54999.99	NULL
2	2	2024-04-02	32999.99	NULL
3	3	2024-04-03	16499.99	NULL
4	4	2024-04-04	10999.99	NULL
5	5	2024-04-05	8799.99	NULL
6	6	2024-04-06	38499.99	NULL
7	7	2024-04-07	60499.99	NULL
9	9	2024-04-09	10999.99	shipped
10	1	2024-04-10	39999.99	NULL

```
9 rows in set (0.00 sec)
```

12. Write an SQL query to calculate and update the number of orders placed by each customer in the "Customers" table based on the data in the "Orders" table.

ALTER TABLE Customers

ADD COLUMN NumOrders INT;

```
mysql> ALTER TABLE Customers
-> ADD COLUMN NumOfOrders INT;
Query OK, 0 rows affected (0.06 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> select * from customers;
```

CustomerID	FirstName	LastName	Email	Phone	Address	NumOfOrders
1	Ravi	Kumar	ravi.kumar@gmail.com	1234567890	123 Main St, Vijayawada	NULL
2	Sarala	Reddy	sarala.reddy@gmail.com	0987654321	456 Main St, Guntur	NULL
3	Prasad	Rao	prasad.rao@gmail.com	9876543210	789 Main St, Visakhapatnam	NULL
4	Anusha	Gowda	anusha.gowda@gmail.com	0123456789	101 Main St, Tirupati	NULL
5	Rajesh	Babu	rajesh.babu@gmail.com	8765432109	222 Main St, Nellore	NULL
6	Shobha	Chowdary	shobha.chowdary@gmail.com	5432109876	333 Main St, Kakinada	NULL
7	Prakash	Naidu	prakash.naidu@gmail.com	6543210987	444 Main St, Rajahmundry	NULL
8	Vijaya	Kumari	vijaya.kumari@gmail.com	2345678901	555 Main St, Amaravati	NULL
9	Ram	Gopal	ram.gopal@gmail.com	4321098765	666 Main St, Vizianagaram	NULL
10	Lakshmi	Devi	example@example.com	3456789012	466 Main St, Vizag	NULL
11	Ram	Reddy	ram@gmail.com	1234123412	123 Main St Vijayawada	NULL

```
11 rows in set (0.00 sec)
```

UPDATE Customers c

SET NumOrders = (

SELECT COUNT(*)

FROM Orders o

WHERE o.CustomerID = c.CustomerID

);

```
mysql> UPDATE Customers c
-> SET NumOfOrders = ( select count(*) from orders o WHERE o.CustomerID = c.CustomerID);
Query OK, 11 rows affected (0.02 sec)
Rows matched: 11 Changed: 11 Warnings: 0

mysql> select * from customers;
```

CustomerID	FirstName	LastName	Email	Phone	Address	NumOfOrders
1	Ravi	Kumar	ravi.kumar@gmail.com	1234567890	123 Main St, Vijayawada	2
2	Sarala	Reddy	sarala.reddy@gmail.com	0987654321	456 Main St, Guntur	1
3	Prasad	Rao	prasad.rao@gmail.com	9876543210	789 Main St, Visakhapatnam	1
4	Anusha	Gowda	anusha.gowda@gmail.com	0123456789	101 Main St, Tirupati	1
5	Rajesh	Babu	rajesh.babu@gmail.com	8765432109	222 Main St, Nellore	1
6	Shobha	Chowdary	shobha.chowdary@gmail.com	5432109876	333 Main St, Kakinada	1
7	Prakash	Naidu	prakash.naidu@gmail.com	6543210987	444 Main St, Rajahmundry	1
8	Vijaya	Kumari	vijaya.kumari@gmail.com	2345678901	555 Main St, Amaravati	0
9	Ram	Gopal	ram.gopal@gmail.com	4321098765	666 Main St, Vizianagaram	1
10	Lakshmi	Devi	example@example.com	3456789012	466 Main St, Vizag	0
11	Ram	Reddy	ram@gmail.com	1234123412	123 Main St Vijayawada	0

```
11 rows in set (0.00 sec)
```

Task 3. Aggregate functions, Having, Order By, GroupBy and Joins:

1. Write an SQL query to retrieve a list of all orders along with customer information (e.g., customer name) for each order.

```
select * from orders o JOIN Customers c ON o.customerID=c.CustomerID;
```

```
mysql> select * from orders o JOIN Customers c ON o.customerID=c.CustomerID;
```

OrderID	CustomerID	OrderDate	TotalAmount	Status	CustomerID	FirstName	LastName	Email	Phone	Address	NumOfOrders
1	1	2024-04-01	54999.99	NULL	1	Ravi	Kumar	ravi.kumar@gmail.com	1234567890	123 Main St, Vijayawada	2
2	2	2024-04-02	32999.99	NULL	2	Sarala	Reddy	sarala.reddy@gmail.com	0987654321	456 Main St, Guntur	1
3	3	2024-04-03	16499.99	NULL	3	Prasad	Rao	prasad.rao@gmail.com	9876543210	789 Main St, Visakhapatnam	1
4	4	2024-04-04	10999.99	NULL	4	Anusha	Gowda	anusha.gowda@gmail.com	0123456789	101 Main St, Tirupati	1
5	5	2024-04-05	8799.99	NULL	5	Rajesh	Babu	rajesh.babu@gmail.com	8765432109	222 Main St, Nellore	1
6	6	2024-04-06	38499.99	NULL	6	Shobha	Chowdary	shobha.chowdary@gmail.com	5432109876	333 Main St, Kakinada	1
7	7	2024-04-07	60499.99	NULL	7	Prakash	Naidu	prakash.naidu@gmail.com	6543210987	444 Main St, Rajahmundry	1
9	9	2024-04-09	10999.99	shipped	9	Ram	Gopal	ram.gopal@gmail.com	4321098765	666 Main St, Vizianagaram	1
10	1	2024-04-10	39999.99	NULL	1	Ravi	Kumar	ravi.kumar@gmail.com	1234567890	123 Main St, Vijayawada	2

```
9 rows in set (0.01 sec)
```

2. Write an SQL query to find the total revenue generated by each electronic gadget product. Include the product name and the total revenue.

```
select p.productname, SUM(p.price*od.quantity) as totalRevenue
```

```
from products p
```

```
JOIN orderdetails od ON p.productID=od.productID
```

```
GROUP BY p.productname;
```

```
mysql> select p.productname, SUM(p.price*od.quantity) as totalRevenue
-> from products p
-> JOIN orderdetails od ON p.productID=od.productID
-> GROUP BY p.productname;
```

productname	totalRevenue
HP Laptop	54999.99
Smartphone	32999.99
Tablet	16499.99
Headphones	10999.99
Smartwatch	8799.99
Camera	38499.99
TV	60499.99
Printer	10999.99

```
8 rows in set (0.01 sec)
```

3. Write an SQL query to list all customers who have made at least one purchase. Include their names and contact information.

```
select distinct group_concat(c.FirstName , ' ', c.LastName) as Name, c.Address, c.phone,
c.email
```

```
FROM Customers c
```

```
JOIN Orders o ON o.CustomerID=c.CustomerID
```

```
GROUP BY c.CustomerID;
```

```
mysql> select distinct group_concat(c.FirstName, ' ', c.LastName) as Name, c.Address, c.phone, c.email
-> FROM Customers c
-> JOIN Orders o ON o.CustomerID=c.CustomerID
-> GROUP BY c.CustomerID;
```

Name	Address	phone	email
Anusha Gowda	101 Main St, Tirupati	0123456789	anusha.gowda@gmail.com
Prakash Naidu	444 Main St, Rajahmundry	6543210987	prakash.naidu@gmail.com
Prasad Rao	789 Main St, Visakhapatnam	9876543210	prasad.rao@gmail.com
Rajesh Babu	222 Main St, Nellore	8765432109	rajesh.babu@gmail.com
Ram Gopal	666 Main St, Vizianagaram	4321098765	ram.gopal@gmail.com
Ravi Kumar,Ravi Kumar	123 Main St, Vijayawada	1234567890	ravi.kumar@gmail.com
Sarala Reddy	456 Main St, Guntur	0987654321	sarala.reddy@gmail.com
Shobha Chowdary	333 Main St, Kakinada	5432109876	shobha.chowdary@gmail.com

8 rows in set (0.01 sec)

4. Write an SQL query to find the most popular electronic gadget, which is the one with the highest total quantity ordered. Include the product name and the total quantity ordered.

```
select p.productName, SUM(od.quantity) as TotalQuantity
```

```
from products p
```

```
JOIN OrderDetails od ON od.ProductID=p.productID
```

```
GROUP BY p.productID
```

```
ORDER BY TotalQuantity DESC LIMIT 1;
```

```
mysql> select p.productName, SUM(od.quantity) as TotalQuantity
-> from products p
-> JOIN OrderDetails od ON od.ProductID=p.productID
-> GROUP BY p.productID
-> ORDER BY TotalQuantity DESC LIMIT 1;
```

productName	TotalQuantity
HP Laptop	1

1 row in set (0.00 sec)

5. Write an SQL query to retrieve a list of electronic gadgets along with their corresponding categories.

Adding category column to the Products table

```
ALTER TABLE Products
```

```
ADD COLUMN Category VARCHAR(50);
```

Updating the existing rows

```
UPDATE Products
```

```
SET Category =
```

```
CASE ProductID

WHEN 1 THEN 'Laptop'

WHEN 2 THEN 'Mobile'

WHEN 3 THEN 'Tablet'

WHEN 4 THEN 'Headphones'

WHEN 5 THEN 'Smartwatch'

WHEN 6 THEN 'Camera'

WHEN 7 THEN 'TV'

WHEN 8 THEN 'Gaming'

WHEN 9 THEN 'Printer'

WHEN 10 THEN 'Router'

WHEN 11 THEN 'Laptop'

END;
```

```
mysql> ALTER TABLE Products
-> ADD COLUMN Category VARCHAR(50);
Query OK, 0 rows affected (0.10 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> UPDATE Products
-> SET Category =
-> CASE ProductID
-> WHEN 1 THEN 'Laptop'
-> WHEN 2 THEN 'Mobile'
-> WHEN 3 THEN 'Tablet'
-> WHEN 4 THEN 'Headphones'
-> WHEN 5 THEN 'Smartwatch'
-> WHEN 6 THEN 'Camera'
-> WHEN 7 THEN 'TV'
-> WHEN 8 THEN 'Gaming'
-> WHEN 9 THEN 'Printer'
-> WHEN 10 THEN 'Router'
-> WHEN 11 THEN 'Laptop'
-> END;
Query OK, 1 row affected (0.01 sec)
Rows matched: 11 Changed: 1 Warnings: 0
```

```
select * from Products;
```

```
mysql> select * from Products;
```

ProductID	ProductName	Description	Price	Category
1	HP Laptop	High performance laptop	54999.99	Laptop
2	Smartphone	Latest smartphone model	32999.99	Mobile
3	Tablet	Compact tablet	16499.99	Tablet
4	Headphones	Wireless headphones	10999.99	Headphones
5	Smartwatch	Fitness smartwatch	8799.99	Smartwatch
6	Camera	DSLR camera	38499.99	Camera
7	TV	Smart LED TV	60499.99	TV
8	Gaming Console	Next-gen gaming console	21999.99	Gaming
9	Printer	Wireless printer	10999.99	Printer
10	Router	High-speed router	5499.99	Router
11	Dell Laptop	High performance laptop	49999.99	Laptop

```
11 rows in set (0.00 sec)
```

6. Write an SQL query to calculate the average order value for each customer. Include the customer's name and their average order value.

```
Select group_concat(c.FirstName , ' ', c.LastName) as Name, AVG(o.totalAmount)
From Customers c
JOIN Orders o ON o.CustomerID=c.CustomerID
GROUP BY c.CustomerID;
```

```
mysql> Select group_concat(c.FirstName , ' ', c.LastName) as Name, AVG(o.totalAmount)
-> From Customers c
-> JOIN Orders o ON o.CustomerID=c.CustomerID
-> GROUP BY c.CustomerID;
```

Name	AVG(o.totalAmount)
Ravi Kumar,Ravi Kumar	47499.990000
Sarala Reddy	32999.990000
Prasad Rao	16499.990000
Anusha Gowda	10999.990000
Rajesh Babu	8799.990000
Shobha Chowdary	38499.990000
Prakash Naidu	60499.990000
Ram Gopal	10999.990000

```
8 rows in set (0.00 sec)
```

7. Write an SQL query to find the order with the highest total revenue. Include the order ID, customer information, and the total revenue.

```
Select o.orderID, group_concat(c.FirstName , ' ', c.LastName) as Name, c.phone, c.email,
c.Address, o.TotalAmount
From orders o
JOIN Customers c ON o.CustomerID=c.CustomerID
```

GROUP BY o.OrderID

ORDER BY o.TotalAmount DESC LIMIT 1;

```
mysql> Select o.orderID, group_concat(c.FirstName , ' ', c.LastName) as Name, c.phone, c.email, c.Address, o.TotalAmount
-> From orders o
-> JOIN Customers c ON o.CustomerID=c.CustomerID
-> GROUP BY o.OrderID
-> ORDER BY o.TotalAmount DESC LIMIT 1;
```

orderID	Name	phone	email	Address	TotalAmount
7	Prakash Naidu	6543210987	prakash.naidu@gmail.com	444 Main St, Rajahmundry	60499.99

1 row in set (0.01 sec)

8. Write an SQL query to list electronic gadgets and the number of times each product has been ordered.

Select p.productName, SUM(od.quantity) as NumOfTimesOrdered

From Products p

JOIN OrderDetails od ON od.productID=p.productID

GROUP BY p.productID;

```
mysql> Select p.productName, SUM(od.quantity) as NumOfTimesOrdered
-> From Products p
-> JOIN OrderDetails od ON od.productID=p.productID
-> GROUP BY p.productID;
```

productName	NumOfTimesOrdered
HP Laptop	1
Smartphone	1
Tablet	1
Headphones	1
Smartwatch	1
Camera	1
TV	1
Printer	1

8 rows in set (0.01 sec)

9. Write an SQL query to find customers who have purchased a specific electronic gadget product. Allow users to input the product name as a parameter.

SET @gadget = 'Laptop';

```
mysql> SET @gadget = 'HP Laptop';
Query OK, 0 rows affected (0.01 sec)
```

SELECT c.CustomerID, GROUP_CONCAT(c.FirstName, ' ', c.LastName) as Name,
p.ProductName

```

FROM Customers c

JOIN Orders o ON c.CustomerID = o.CustomerID

JOIN OrderDetails od ON o.OrderID = od.OrderID

JOIN Products p ON p.ProductID = od.ProductID

WHERE p.ProductName LIKE CONCAT ('%', @gadget, '%')

GROUP BY c.CustomerID, p.ProductName;

```

```

mysql> SELECT c.CustomerID, GROUP_CONCAT(c.FirstName, ' ', c.LastName) as Name, p.ProductName
-> FROM Customers c
-> JOIN Orders o ON c.CustomerID = o.CustomerID
-> JOIN OrderDetails od ON o.OrderID = od.OrderID
-> JOIN Products p ON p.ProductID = od.ProductID
-> WHERE p.ProductName LIKE CONCAT ('%', @gadget, '%')
-> GROUP BY c.CustomerID, p.ProductName;
+-----+-----+-----+
| CustomerID | Name      | ProductName |
+-----+-----+-----+
|          1 | Ravi Kumar | HP Laptop   |
+-----+-----+-----+
1 row in set (0.02 sec)

```

10. Write an SQL query to calculate the total revenue generated by all orders placed within a specific time period. Allow users to input the start and end dates as parameters.

```

SET @start_date = '2024-01-01';

SET @end_date = '2024-11-12';

Select SUM(o.TotalAmount) as TotalRevenue

From Orders o

WHERE o.orderDate BETWEEN @start_date AND @end_date;

```

```

mysql> SET @start_date = '2024-01-01';
Query OK, 0 rows affected (0.00 sec)

mysql> SET @end_date = '2024-11-12';
Query OK, 0 rows affected (0.00 sec)

mysql> Select SUM(o.TotalAmount) as TotalRevenue
-> From Orders o
-> WHERE o.orderDate BETWEEN @start_date AND @end_date;
+-----+
| TotalRevenue |
+-----+
|    274299.91 |
+-----+
1 row in set (0.01 sec)

```


Task 4. Subquery and its type:

1. Write an SQL query to find out which customers have not placed any orders.

```
Select customerID, group_concat(firstName, ' ', lastName) as Name from Customers  
WHERE CustomerID NOT IN (select DISTINCT customerID from Orders)  
GROUP BY CustomerID;
```

```
mysql> Select customerID, group_concat(firstName, ' ', lastName) as Name from Customers  
-> WHERE CustomerID NOT IN (select DISTINCT customerID from Orders)  
-> GROUP BY CustomerID;  
+-----+  
| customerID | Name           |  
+-----+  
|          8 | Vijaya Kumari |  
|         10 | Lakshmi Devi  |  
|         11 | Ram Reddy     |  
+-----+  
3 rows in set (0.01 sec)
```

2. Write an SQL query to find the total number of products available for sale.

```
Select COUNT(*) as TotalNumberOfProducts from Products;
```

```
mysql> Select COUNT(*) as TotalNumberOfProducts from Products;  
+-----+  
| TotalNumberOfProducts |  
+-----+  
|                   11 |  
+-----+  
1 row in set (0.01 sec)
```

3. Write an SQL query to calculate the total revenue generated by TechShop.

```
Select SUM(TotalAmount) as Total_Revenue from Orders;
```

```
mysql> Select SUM(TotalAmount) as Total_Revenue from Orders;  
+-----+  
| Total_Revenue |  
+-----+  
|    274299.91 |  
+-----+  
1 row in set (0.01 sec)
```

4. Write an SQL query to calculate the average quantity ordered for products in a specific category. Allow users to input the category name as a parameter.

```
SET @category_name = 'Laptop';
```

```

SELECT p.category, AVG(od.Quantity) AS AverageQuantityOrdered
FROM OrderDetails od
JOIN Products p ON od.ProductID = p.ProductID
WHERE p.Category = @category_name
GROUP BY p.category;

```

```

mysql> SELECT p.category, AVG(od.Quantity) AS AverageQuantityOrdered
-> FROM OrderDetails od
-> JOIN Products p ON od.ProductID = p.ProductID
-> WHERE p.Category = @category_name
-> GROUP BY p.category;
+-----+-----+
| category | AverageQuantityOrdered |
+-----+-----+
| Laptop   | 1.0000                  |
+-----+-----+
1 row in set (0.00 sec)

```

5. Write an SQL query to calculate the total revenue generated by a specific customer. Allow users to input the customer ID as a parameter.

```
SET @customer_id = '6';
```

```
Select o.CustomerID, group_concat(firstName, ' ', lastName) as Name, SUM(o.TotalAmount)
as Revenue_Generated
```

```
From Orders o
```

```
JOIN Customers c ON c.CustomerID = o.CustomerID
```

```
WHERE o.CustomerID = @customer_id
```

```
GROUP BY CustomerID;
```

```

mysql> Select o.CustomerID, group_concat(firstName, ' ', lastName) as Name, SUM(o.TotalAmount) as Revenue_Generated
-> From Orders o
-> JOIN Customers c ON c.CustomerID = o.CustomerID
-> WHERE o.CustomerID = @customer_id
-> GROUP BY CustomerID;
+-----+-----+-----+
| CustomerID | Name           | Revenue_Generated |
+-----+-----+-----+
| 6          | Shobha Chowdary | 38499.99          |
+-----+-----+-----+
1 row in set (0.00 sec)

```

(or)

```

SELECT CustomerID, (SELECT SUM(TotalAmount) FROM Orders WHERE CustomerID =
@customer_id) as Revenue_Generated

```

FROM Orders

WHERE CustomerID = @customer_id;

```
mysql> SELECT CustomerID, (SELECT SUM(TotalAmount) FROM Orders WHERE CustomerID = @customer_id) as Revenue_Generated
-> FROM Orders
-> WHERE CustomerID = @customer_id;
+-----+-----+
| CustomerID | Revenue_Generated |
+-----+-----+
|          6 |          38499.99 |
+-----+-----+
1 row in set (0.00 sec)
```

6. Write an SQL query to find the customers who have placed the most orders. List their names and the number of orders they've placed.

SELECT CustomerID, group_concat(firstName, ' ', lastName) as Name,

(SELECT COUNT(OrderByID) FROM Orders o WHERE o.CustomerID = c.CustomerID) as
Num_of_Orders_Placed

FROM Customers c

GROUP BY CustomerID

HAVING Num_of_Orders_Placed = (SELECT COUNT(OrderByID) from Orders o

GROUP BY CustomerID

ORDER BY COUNT(OrderByID) DESC LIMIT 1)

ORDER BY Num_of_Orders_Placed DESC;

```
mysql> SELECT CustomerID, group_concat(firstName, ' ', lastName) as Name,
-> (SELECT COUNT(OrderByID) FROM Orders o WHERE o.CustomerID = c.CustomerID) as Num_of_Orders_Placed
-> FROM Customers c
-> GROUP BY CustomerID
-> HAVING Num_of_Orders_Placed = ( SELECT COUNT(OrderByID) from Orders o
->                                GROUP BY CustomerID
->                                ORDER BY COUNT(OrderByID) DESC LIMIT 1)
-> ORDER BY Num_of_Orders_Placed DESC;
+-----+-----+-----+
| CustomerID | Name      | Num_of_Orders_Placed |
+-----+-----+-----+
|          1 | Ravi Kumar |                2 |
+-----+-----+-----+
1 row in set (0.00 sec)
```

7. Write an SQL query to find the most popular product category, which is the one with the highest total quantity ordered across all orders.

SELECT p.Category as Most_Popular_Category, SUM(od.Quantity) as
Total_Quantity_Ordered

FROM OrderDetails od

JOIN Products p ON od.ProductID = p.ProductID

GROUP BY p.Category

ORDER BY Total_Quantity_Ordered DESC LIMIT 1;

```
mysql> SELECT p.Category AS Most_Popular_Category, SUM(od.Quantity) AS Total_Quantity_Ordered
-> FROM OrderDetails od
-> JOIN Products p ON od.ProductID = p.ProductID
-> GROUP BY p.Category
-> ORDER BY Total_Quantity_Ordered DESC LIMIT 1;
+-----+-----+
| Most_Popular_Category | Total_Quantity_Ordered |
+-----+-----+
| Laptop                | 1                      |
+-----+-----+
1 row in set (0.02 sec)
```

8. Write an SQL query to find the customer who has spent the most money (highest total revenue) on electronic gadgets. List their name and total spending.

Select CustomerID, group_concat(firstName, ' ', lastName) as Name,

(SELECT SUM(TotalAmount) from Orders o WHERE o.CustomerID=c.CustomerID)

As Total_Revenue

From Customers c

GROUP BY CustomerID

ORDER BY Total_Revenue DESC LIMIT 1;

```
mysql> Select CustomerID, group_concat(firstName, ' ', lastName) as Name,
-> (SELECT SUM(TotalAmount) from Orders o WHERE o.CustomerID=c.CustomerID)
-> As Total_Revenue
-> From Customers c
-> GROUP BY CustomerID
-> ORDER BY Total_Revenue DESC LIMIT 1;
+-----+-----+-----+
| CustomerID | Name       | Total_Revenue |
+-----+-----+-----+
| 1          | Ravi Kumar | 94999.98      |
+-----+-----+-----+
1 row in set (0.00 sec)
```

9. Write an SQL query to calculate the average order value (total revenue divided by the number of orders) for all customers.

Select CustomerID, group_concat(firstName, ' ', lastName) as Name,

(SELECT AVG(TotalAmount) from Orders o WHERE o.CustomerID=c.CustomerID)

As Average_Order_Value

From Customers c

GROUP BY CustomerID;

```
mysql> Select CustomerID, group_concat(firstName, ' ', lastName) as Name,  
-> (SELECT AVG(TotalAmount) from Orders o WHERE o.CustomerID=c.CustomerID)  
-> As Average_Order_Value  
-> From Customers c  
-> GROUP BY CustomerID;
```

CustomerID	Name	Average_Order_Value
1	Ravi Kumar	47499.990000
2	Sarala Reddy	32999.990000
3	Prasad Rao	16499.990000
4	Anusha Gowda	10999.990000
5	Rajesh Babu	8799.990000
6	Shobha Chowdary	38499.990000
7	Prakash Naidu	60499.990000
8	Vijaya Kumari	NULL
9	Ram Gopal	10999.990000
10	Lakshmi Devi	NULL
11	Ram Reddy	NULL

11 rows in set (0.00 sec)

10. Write an SQL query to find the total number of orders placed by each customer and list their names along with the order count.

Select CustomerID, group_concat(firstName, ' ', lastName) as Name,

(SELECT COUNT(OrderID) from Orders o WHERE o.CustomerID=c.CustomerID)

As Total_Orders_Placed

From Customers c

GROUP BY CustomerID;

```
mysql> Select CustomerID, group_concat(firstName, ' ', lastName) as Name,  
-> (SELECT COUNT(OrderID) from Orders o WHERE o.CustomerID=c.CustomerID)  
-> As Total_Orders_Placed  
-> From Customers c  
-> GROUP BY CustomerID;
```

CustomerID	Name	Total_Orders_Placed
1	Ravi Kumar	2
2	Sarala Reddy	1
3	Prasad Rao	1
4	Anusha Gowda	1
5	Rajesh Babu	1
6	Shobha Chowdary	1
7	Prakash Naidu	1
8	Vijaya Kumari	0
9	Ram Gopal	1
10	Lakshmi Devi	0
11	Ram Reddy	0

11 rows in set (0.00 sec)